PAPER

Traffic engineering and traffic monitoring in the case of incomplete information

Kodai SATAKE†, **Tatsuya OTOSHI**†, **Yuichi OHSITA**†, ***and*** **Masayuki MURATA**†,

SUMMARY Traffic engineering refers to techniques to accommodate traffic efficiently by dynamically configuring traffic routes so as to adjust to changes in traffic. If traffic changes frequently and drastically, the interval to reconfigure routes should be short. However, with shorter intervals, obtaining traffic information is problematic. To calculate a suitable route, traffic information must be accurately measured for the whole network, and this is difficult in short intervals, owing to the overhead incurred to monitor and collect traffic information. In this paper, we propose a framework for traffic engineering in cases where only partial traffic information can be obtained during each time slot. The proposed framework was inspired by the human brain, and uses conditional probability to make decisions. In this framework, a controller is deployed to (1) obtain a limited amount of traffic information, (2) estimate and predict the probability distribution of the traffic, (3) configure routes considering the probability distribution of future predicted traffic, and (4) select traffic that should be monitored during the next period considering the performance of the route reconfiguration. We evaluated our framework with a simulation. The results demonstrate that our framework improves the efficiency of traffic even when only partial traffic information is monitored during each time slot.
key words: *Traffic engineering, traffic monitoring, incomplete information, Bayesian Brain*

1. Introduction

Traffic engineering (TE) refers to techniques to configure traffic networks to accommodate traffic efficiently, by adjusting to changes in traffic [3]–[14]. In TE methods, a controller periodically collects traffic information, and changes the routes of the traffic flow within the network based on this information. By dynamically reconfiguring the routes, the controller avoids congestion even when traffic changes occur.

Conventional TE methods consider daily traffic changes, and their control intervals are set to one hour or longer. However, if the traffic changes frequently and drastically, the control interval should be shorter. Indeed, when changes result in traffic congestion, this congestion cannot be mitigated until the next time slot. Benson et al. demonstrated that routes should be reconfigured every five seconds in a network where traffic changes frequently and drastically, such as in a datacenter network [15].

TE methods require traffic information to calculate a suitable route during each time slot. If we set a short control interval, traffic information should also be obtained at short intervals. However, in a large network, measuring traffic information accurately for the whole network is difficult in short intervals owing to the overhead incurred to monitor and collect traffic information. As such, we consider a case when only partial traffic information can be monitored and collected during an interval.

Methods to estimate traffic in the whole network from partial traffic information have been proposed [16], [17]. With such methods, the traffic information of the flow that is not included in the collected traffic information is estimated from the collected traffic information, considering the spatial and temporal properties of the traffic. However, estimated traffic includes estimation errors, and these errors affect the performance of TE. To mitigate the impact of estimation errors, TE methods be designed in consideration of estimation errors. On the other hand, traffic monitoring and estimation methods can avoid estimation errors affecting TE by improving the accuracy of the estimation of traffic flow that is important to TE. That is, the TE and traffic monitoring should work together.

We propose a framework in which TE and traffic monitoring cooperate to handle a situation where only partial information is obtained at each time slot. This framework was inspired by the process by which human brains make decisions based on uncertain and incomplete information. A human brain makes many good decisions even in a highly uncertain environment. One promising theoretical model to explain how a human brain makes decisions is the Bayesian decision-making model [18]. According to this model, the human brain has stochastic variables, and it updates these variables with Bayesian estimations every time a new observation is obtained. Then, decisions are made based on these stochastic variables. By doing so, a human brain can make decisions even when only uncertain and incomplete information can be obtained. We applied this process to TE. Specifically, we applied the feature that a human brain increases confidence by repeating Bayesian estimations. In our framework, the controller has stochastic variables regarding traffic, and it updates them every time new traffic information is obtained. By repeating the above steps, the controller understands the probability distributions of traffic, even when only a part of the information is obtained at each time slot.

Manuscript received January 1, 2015.
Manuscript revised January 1, 2015.
†The authors are affiliated with the Graduate School of Information Science and Technology, Osaka University.
Part of the research results were achieved under the program “Research and Development of Innovative Network Technologies to Create the Future," with the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.
Part of this paper was presented at ICTC 2016 [1] and ITNAC 2017 [2].
DOI: 10.1587/trans.E0.??.1

Copyright © 200x The Institute of Electronics, Information and Communication Engineers

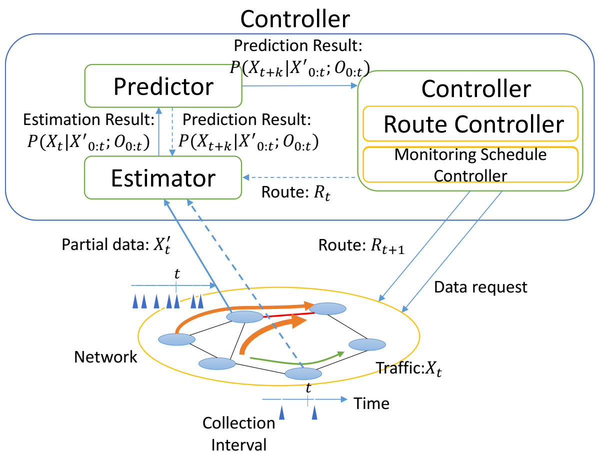

Fig. 1 Overview of our framework

The controller changes the routes based on the probability distributions of the traffic. In addition, the controller decides the points to be monitored in the next time slot based on the probability distribution of the traffic.

The remaining paper is organized as follows. Section 2 provides an overview of our framework. Section 3 describes the details of each step of our framework. Section 4 presents an evaluation of our method. Section 5 describes an acceleration method for our method and presents an evaluation of its calculation time. Section 6 presents our concluding remarks.

2. Framework for Traffic Engineering under Uncertain Traffic Information

Figure 1 shows an overview of our framework. In our framework, a controller is deployed. The controller (1) obtains a limited amount of traffic information, (2) estimates and predicts the probability distribution of the traffic, (3) configures the routes considering the probability distribution of future predicted traffic, and (4) configures the traffic to be monitored during the next period, in consideration of the performance of the route reconfiguration. To perform the above operations based on Bayes ' decision-making theory, the controller includes the following modules: an estimator, predictor, route controller, and monitoring schedule controller. We explain each of these modules in the following section.

In this work, we denote the traffic rates of all flows during time slot t by X_t and the traffic rates of flow f at time slot t by $x_{t,f}$. First, the controller observes part of the traffic. We denote the traffic observed at time slot t by X'_t, and the element of X'_t corresponding to the flow f by $x'_{t,f}$. We also denote the route at time slot t by R_t, and the set of the observed flows at time slot t by O_t. Thus, $X_{t:t+k}$ is $(X_t, X_{t+1}, \ldots, X_{t+k})$. Similarly, we define $X'_{t:t+k} = (X'_t, X'_{t+1}, \ldots, X'_{t+k})$ and $O_{t:t+k} = (O_t, O_{t+1}, \ldots, O_{t+k})$

2.1 Estimator

The estimator estimates the amount of traffic in the current network from partially observed traffic information. In our framework, it estimates the posterior distribution of traffic volume X_t after obtaining the partially monitored traffic X'_t.

The posterior distribution of traffic volume X_t is calculated by

$$P(X_t|X'_t; O_t) = \frac{1}{P\,(X'_t|O(t))} P(X'_t|X_t; O_t) P(X_t) \quad (1)$$

where $P(X_t)$ is a prior distribution of X_t. The traffic volume predicted by the predictor at time slot $t-1$ can be used as the prior distribution. That is, the posterior distribution of the traffic volume X_t is calculated by

$$P(X_t|X'_{0:t}; O_{0:t}) = \frac{1}{P(X'_t)} P(X'_t|X_t; O_t) P(X_t|X'_{0:t-1}; O_{0:t-1}) \quad (2)$$

where $P(X_t|X'_{0:t-1}; O_{0:t-1})$ is the probability distribution of X_t predicted at time slot $t-1$. By using the traffic predicted during previous time slots, the traffic volume for the entire network can be estimated even when only limited traffic is monitored.

2.2 Predictor

The predictor predicts the probability distribution of future traffic volume based on past traffic. That is, it predicts $P(X_{t+k}|X'_{0:t}; O_{0:t})$.

The predictor uses a prediction model with parameter θ to predict traffic rates after time slot $t+1$ from past traffic rates by $P(X_{t+1:t+n}|X_{t-m:t}; \theta)$, where m is the length of the past traffic rates used by the predictor, and n is the length of the time slot that can be predicted.

By using this prediction model, future traffic rates are predicted from previously monitored traffic:

$$\begin{aligned} &P(X_{t+1:t+n}|X'_{0:t};O_{0:t},\theta)= \\ &\sum_{X_{t-m:t}} P(X_{t+1:t+n}|X_{t-m:t};\theta) P(X_{t-m:t}|X'_{0:t}; O_{0:t}) \end{aligned} \quad (3)$$

Then, $P(X_{t+k}|X'_{0:t}; O_{0:t}, \theta)$ can be obtained by

$$P(X_{t+k}|X'_{0:t}; O_{0:t}, \theta) = \sum_{X_{t:t+k-1}, X_{t+k+1:t+n}} P(X_{t+1:t+n}|X'_{0:t}; O_{0:t}, \theta). \quad (4)$$

2.3 Route controller

The route controller calculates routes to achieve the required performance considering the predicted probability distribution of traffic rates.

Previously, we proposed a method called stochastic MP-TE, which configures routes considering the predicted probability of future traffic [19]. With stochastic MP-TE, routes are calculated so as to minimize the weighted sum of a cost function indicating the network performance and the cost of changing routes under the constraint that the probability that traffic passing a link should not exceed a threshold. In the framework proposed here, the route controller is based on stochastic MP-TE, and it calculates routes by solving the following optimization problem:

$$minimize : E\left[\sum_{i=t+1}^{t+h} \{(1-w)f(X_i, R_i) + w||R_i - R_{i-1}||^2\}\right] \tag{5}$$

$$s.t. : P\left(y_i^l(X_i, R_i) > c^l\right) \leq p \tag{6}$$

where $f(X_i, R_i)$ is a cost function indicating the network performance, $||R_i - R_{i-1}||^2$ is the cost of changing routes from R_{i-1} to R_i, and $y_i^l(X_i, R_i)$ is the amount of traffic passing link l when the traffic rate is X_i and the routes are set to R_i. Further, h is the length of the predictive time series considered by the route controller, w is the weight assigned the cost of changing routes, c^l is the threshold for the amount of traffic passing link i, and p is the acceptable probability that the traffic will exceed the threshold.

Although the routes $R_{k+1}, \ldots, R_{k+h}$ are obtained by solving the above optimization problem, the route controller actually sets R_{k+1} to the network. After collecting data during subsequent time slots, future routes are recalculated with the new prediction results. By doing so, the route controller adaptively corrects the route, even if the predictive distribution is temporally wrong.

2.4 Monitoring schedule controller

In the proposed framework, the monitoring schedule controller decides on what traffic will be monitored during the next time slot, in order to minimize the expectation value of the cost after the route changes based on the traffic monitored during the next time slot.

The monitoring schedule controller decides on subsequent traffic monitoring by solving the following optimization problem:

$$minimize : E_{P(X_{t+1})P(X'_t|O_t)}\left[f\left(X_{t+1}, R_{k+1}(X'_t, O_t)\right)\right] \tag{7}$$

$$s.t. C(O_t) \leq W \tag{8}$$

where $E_{P(X)}[f(X)]$ is the expectation value of $f(X)$ under the probability distribution $P(X)$, and $R_{t+1}(X'_t, O_t)$ is the route configuration for time slot $t+1$ calculated by the route controller when X'_t is observed at time slot t by monitoring the flows included in O_t. Further, $C(O_t)$ is the overhead required to monitor the traffic in O_t, W is the acceptable overhead, and $P(X'_t|O_t) = \sum_{X_t} P(X'_t|X_t; O_t), where P(X_t)$ is the probability of the observed traffic X_t when the traffic in O_t is monitored. When solving the above optimization problem, the probability distributions $P(X_t)$ and $P(X_{t+1})$ are unknown. In this framework, we use the predicted probability distributions $P(X_t|X'_{0:t-1}; O_{0:t-1})$ and $P(X_{t+1}|X'_{0:t-1}; O_{0:t-1})$ in place of these distributions.

3. Dynamic traffic engineering and traffic monitoring based on the framework

In this section, we describe dynamic TE and traffic monitoring based on our framework. We call this method the *stochastic control considering uncertainty* (*SCCU*). Our framework comprises the estimator, predictor, route controller and monitoring schedule controller. In this section, we specify each of these modules.

3.1 Estimator

The estimator estimates $P(X_t|X'_{0:t}; O_{0:t})$. To estimate $P(X_t|X'_{0:t}; O_{0:t})$, we need to define $P(X'_t|X_t; O_t)$, which indicates the probability distribution of the observed traffic.

We assume that monitoring O_t does not provide any information about the traffic rates of the flow $x_{t,f}$ in cases where the flow f is not included in O_t. If the flow f is included in O_t, it can be observed accurately as follows: $P(X'_t|X_t; O_t)$ is

$$P(x'_{t,f}|X_t; O_t) = \begin{cases} \delta(x'_{t,f} - x_{t,f}) & (f \in O_t) \\ \mathcal{U}(0, \infty) & \text{(otherwise)} \end{cases} \tag{9}$$

where $\delta(x)$ is the Dirac delta function, and $\mathcal{U}(a, b)$ is a uniform distribution between a and b.

By using Eq. (9), $P(x_t|X'_{0:t}; O_{0:t})$ is estimated by

$$P(x_t|X'_{0:t}; O_{0:t}) = \begin{cases} \delta(x'_{t,f} - x_{t,f}) & (f \in O_t) \\ P(X_t|X'_{0:t-1}; O_{0:t-1}) & \text{(otherwise)} \end{cases} \tag{10}$$

We approximate $\delta(x'_{t,f} - x_{t,f})$ in Eq. (10) using a Gaussian distribution with very little variance, such that the probability distribution can be handled easily.

3.2 Predictor

The predictor predicts $P(X_{t+k}|X'_{0:t}; O_{0:t})$ by using the model $P(X_{t+1:t+n}|X_{t-m:t}; \theta)$. In this paper, we use the following simple model:

$$x_{t+1,f} = x_{t,f} + \epsilon_{t,f} \tag{11}$$

where $\epsilon_{t,f}$ is Gaussian noise. Indeed, there may be more sophisticated models than this. Nevertheless, by using this model, $P(X_{t+1}|X_t; \theta)$ is obtained by

$$P(x_{t+1,f}|x_{t,f}; \sigma_{t,f}) = \mathcal{N}(x_{t,f}, \sigma^2_{t,f}) \tag{12}$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution whose mean and variance are μ and σ^2, $\sigma^2_{t,f}$ denotes variance in the traffic rates for flow f at time slot t, and $\sigma^2_{t,f}$ is calculated using the last s monitored traffic amounts for flow f.

By using $P(x_{t+1,f}|x_{t,f};\sigma_{t,f})$,

$$P(x_{t+1,f}|X'_{0:t};O_{0:t},\sigma_{t,f}) = \sum_{x_{t,f}} P(x_{t+1,f}|x_{t,f};\sigma_{t,f})P(x_{t,f}|X'_{0:t};O_{0:t}) \tag{13}$$

where $P(x_{t,f}|X'_{0:t};O_{0:t})$ is a probability distribution of the current traffic rates estimated by the estimator. By continuing the following calculation, we can obtain the probability distribution of future traffic rates:

$$P(x_{t+k,f}|X'_{0:t};O_{0:t},\sigma_{t,f}) = \sum_{x_{t+k-1,f}} P(x_{t+k,f}|x_{t+k-1,f};\sigma_{t,f})P(x_{t+k-1,f}|X'_{0:t};O_{0:t},\sigma_{t,f}) \tag{14}$$

Although this model is simple, it captures the following features of traffic prediction:

- The variance in traffic flows with large fluctuations becomes considerable.
- The variance of the predicted traffic rates in the distant future increases.

Therefore, this model can be used to identify flow in which the traffic rate is uncertain, and it can be used by the route controller in consideration of the uncertainty of traffic rates.

3.3 Route controller

The route controller calculates routes such that the required performance is provided, considering the predicted probability distribution of traffic rates.

Multiple routes between the source and destination node pairs are calculated in advance, and the route controller calculates the suitable ratio of the traffic of the flow passing the routes calculated in advance. We define the element R_t such that $R_t^{i,j}$ is the ratio of the traffic of flow j passing route i. We also define a matrix G, whose element $G^{i,j}$ takes the value 1 if route j passes link i, and 0 otherwise. The traffic passing link l at time slot t can be obtained by $\sum_{f,j} G^{l,j} R_t^{j,f} x_{t,f}$.

Insofar as the aim is to avoid congestion, we define the optimization problem solved by the route controller as follows:

$$minimize: \sum_{k=1}^{h} \|R_{t+k} - R_{t+k-1}\| \tag{15}$$

$$subject\ to: \forall 1 \le k \le h, \forall l, P\left[\sum_{f,j} G^{l,j} R^{f,j}_{t+k} x_{t+k,f} > c_l\right] \le p_k \tag{16}$$

$$\forall 1 \le k \le h, \forall i, \forall j, R^{i,j}_{t+k} \in [0,1] \tag{17}$$

$$\forall 1 \le k \sum_{i \in \wp(j)} R^{i,j}_{t+k} = 1 \tag{18}$$

where c_l is the threshold for the traffic passing link l, and p_k is the acceptable probability that the traffic passing link l exceeds this threshold. Further, $P[\sum_{i,l} G^{l,j} R^{f,j}_{t+k} x_{t+k,f} > c_l]$ is the probability that the rates of the traffic passing link l will exceed c_l. This probability is obtained from the probability distribution of the predicted traffic, $P(X_{t+k}|X'_{0:t};O_{0:t})$.

However, optimal solutions are unnecessary, that is, routes without congestion are sufficient. Thus, rather than solving the above optimization problem, we obtain routes by minimizing the following equation:

$$L(R_{t+1:t+h}) = \sum_{k=1}^{h} \|R_{t+k} - R_{t+k-1}\| + \sum_{k=1}^{h}\sum_{l} \lambda_{l,h}\left(P\left[\sum_{j} G^{l,j} R^{f,j}_{t+k} x_{t+k,f} > c_l\right] - p_k\right)^+ + \sum_{k=1}^{h}\sum_{l} \Lambda_{l,h}\left(E_{X_{t+k}}\left[\sum_{j} G^{l,j} R^{f,j}_{t+k} x_{t+k,f}\right] - c_l\right)^+ \tag{19}$$

Here, $(x)^+$ is x, if $x \ge 0$, and otherwise 0. The constraints related to the third term in Eq. (19) are not included in Eqs. (15) and (16), but the third term in Eq. (19) is added in order to accelerate the search for a solution when the predicted link utilization is higher than c_l. Here, $\lambda_{l,h}$ and $\Lambda_{l,h}$ denote the weights to the constraints.

The suitable routes $R_{t+1:t+h}$ are calculated such that $L(R_{t+1:t+h})$ is minimized. We use the steepest decent method to minimize $L(R_{t+1:t+h})$, where the optimal $R_{t+1:t+h}$ is obtained by continuing the following update:

- Update $R_{t+1:t+h}$ so as to make $L(R_{t+1:t+h})$ small.

$$R^{i,j}_{t+k} \leftarrow \left(R^{i,j}_{t+k} - \alpha \frac{\partial L(R_{t+1:t+h})}{\partial R^{i,j}_{t+k}}\right)^+$$

- Scale $R^{i,j}_{t+k}$ to satisfy Eq. (18).

$$R^{i,j}_{t+k} \leftarrow \frac{1}{\sum_n R^{n,j}_{t+k}} R^{i,j}_{t+k}$$

Assuming that the control interval is sufficiently short, we do not need to obtain the optimal solution at each control interval, because the difference between the optimal solution of R_{t+k} and the current routes R_t is small, unless significant traffic changes occur. In addition, even if the current solution is not optimal, a solution that is closer to the optimal solution than the current solution is obtained during the next time slot. Therefore, a small number of iterations suffices for the above update.

3.4 Monitoring schedule controller

The monitoring schedule controller decides which traffic will be monitored. This traffic is determined by selecting nodes to monitor the traffic. We denote the set of nodes by N. We also denote by F_n the set of flows whose traffic rates can be monitored by node n. If n is selected as the node that monitors the traffic at time slot t, all flows in F_n are added to O_t.

The traffic that will be monitored is determined based on the following optimization problem:

$$minimize : L^{\mathrm{opt}}(O_{t+1}) \tag{20}$$

$$subject\ to : C(O_{t+1}) \leq W \tag{21}$$

where $L^{\mathrm{opt}}(O_{t+1})$ is the optimal value of the Lagrange function $L(R_{t+1:t+h})$ defined in the previous section when O_{t+1} is observed, and $C(O_{t+1})$ is the overhead required to observe O_{t+1}.

In this study, we obtain the value of $L^{\mathrm{opt}}(O_{t+1})$, assuming that $x'_{t+1,f}$ for flow f included in O_{t+1} is $E_{P(X_{t+1}|X'_{0:t};O_{0:t})}[X_{t+1,f}]$. To obtain the optimal solutions, a comparison of $L^{\mathrm{opt}}(O_{t+1})$ in all cases of O_{t+1} is required, which incurs considerable calculation time. Therefore, we apply a greedy algorithm.

In addition to selecting the traffic to be monitored based on the above optimization problem, we should consider the interval to monitor flow. If flow f has not been monitored for some time, $P(x_{t,f}|X'_{0:t}, O_{0:t})$ may differ from the actual traffic. Therefore, we set the maximum interval I to monitor the traffic. That is, if there are nodes that were not selected for more than I time slots, we first select them as the nodes that monitor the traffic. Then, we select the remaining monitoring nodes as follows:

1. Set $N - N^{\mathrm{selected}}$ as the candidate nodes that monitor the traffic, where N^{selected} is the set of nodes that were already selected for monitoring traffic.
2. For all $n \in N - N^{\mathrm{selected}}$, calculate O_{t+1} when n is added to the nodes monitoring the traffic, and calculate $C(O_{t+1})$ and $L^{opt}(O_{t+1})$.
3. Select n whose corresponding $L^{opt}(O_{t+1})$ is the smallest among the nodes whose corresponding $C(O_{t+1})$ is less than W.
4. If a node is selected in Step 3, add n selected in Step 3 to N^{selected}, and return to Step 1. Otherwise, end.

After completing the above steps, O_{t+1} is set by adding all flows in F_n for all $n \in N^{\mathrm{selected}}$.

4. Evaluation

4.1 Simulation Environment

4.1.1 Network Topology and Network Traffic

We used the backbone network topology and traffic trace data from Internet2 for our simulation. Internet2 has 9 PoP (Point of Presence) routers, although our method should be evaluated in a larger network that includes more candidate monitoring nodes. Therefore, we connected three access routers to each PoP router. That is, the topology used in our evaluation had 27 access routers and 9 PoP routers, as shown in Figure 2. Each flow was generated between each pair of access routers. We assigned the flows included in the traffic trace data to pairs of access routers by setting the range of IP addresses connected to each access router. In our evaluation, we considered the case where only the access routers can monitor traffic sent from the access routers. We set c_l to 2.5 Gbps for all links in order to evaluate the case where congestion can occur without dynamically reconfiguring the routes.

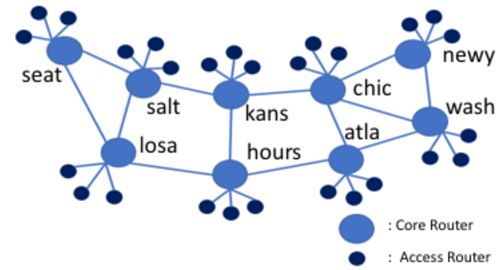

Fig. 2 Expanded Internet2

We used traffic trace data monitored from 0700 h on November 13, 2011 to 1000 h on November 13, 2011. Traffic data was collected by the Netflow protocol at each of the PoP routers. The sampling rate was 1 out of every 100 packets, and aggregated data was exported every 5 min. Although the traffic data did not include information regarding traffic rates whose granularity was less than 5 min, the traffic data did include the start and end times of each flow and the traffic amount of the flow. For our evaluation, we generated the traffic rate of each flow, assuming that it was constant from the start time to the end time.

4.1.2 Parameter Settings

For the evaluation, we ignored $|R_{t+1} - R_t|$ in Eq. (19), and set $\lambda_{l,h}$ and $\Lambda_{l,h}$ to 1 in order to focus on the link utilization achieved by our method. We set h to 1, to focus on the performance during the next time slot.

Moreover, we set p_k to 0.10, I to 220 s, and $\alpha = 0.1$. Further, we set the number of iterations to minimize $L(R_{t+1})$ to 15, and we set the control interval to 10 s. For simplicity, we set the overhead required to observe traffic at a monitoring node to 1.

4.1.3 Metric

We evaluated the rate of links where congestion occurs, because the aim of TE in this paper is to avoid congestion. The rate M of links where congestion occurs is defined by

$$M = \frac{\sum_{t,l} g(y_t^l)}{TL} \tag{22}$$

$$g(y_t^l) = \begin{cases} 1 & (y_t^l(X_t, R_t) > c^l) \\ 0 & \text{(otherwise)} \end{cases} \tag{23}$$

where $y_i^l(X_i, R_i)$ is the amount of traffic passing link l when the traffic rate is X_i and the routes are set to R_i, c^l is the threshold for the amount of traffic passing link i, L is the number of links, and T is the number of time slots. When calculating M, we ignored the first hour in order to omit results that did not have a sufficient amount of monitored traffic to calculate σ_f and affect M.

4.1.4 Comparison

In our evaluation, we compared the proposed SCCU with the following methods.

(1) Control Based on Expected Rate (CBER)

This method calculates routes without considering the probability distribution of traffic, by performing the following steps at each time slot: 1) the controller collects the traffic information from randomly selected monitoring nodes; 2) it predicts the traffic rate in the same manner as SCCU; and 3) calculates the routes by using the values for the traffic rate expected during the next time slot. The routes are calculated so as to minimize the amount of traffic on each link exceeding a threshold $c_l - \Delta c$. That is, the controller solves the following problems:

$$minimize: \sum_l \left(E_{X_{t+k}} \left[\sum_j G^{l,j} R_{t+k}^{f,j} x_{t+k,f} \right] - (c_l - \Delta c) \right)^+ \tag{24}$$

where $(x)^+$ is x if $x \geq 0$, and otherwise 0. With this method, setting Δc to a large value ensures that the amount of traffic on each link is small. In our evaluation, we set Δc to multiple values. By comparing our method with CBER, we intend to demonstrate the impact of considering the probability distribution of traffic volume.

(2) Stochastic Control with Random Select (SCRS)

This method involves the controller randomly selecting nodes to monitor traffic during each time slot. Then, traffic is predicted, and routes are calculated in the same manner as SCCU. A comparison with this method is intended to demonstrate the effect of selecting nodes to monitor traffic at each time slot in consideration of performance.

(3) Long Term Control (LTC)

This method does not change routes until the traffic information is obtained for all flows. With LTC, the controller simply calculates the routes so as to minimize the amount of traffic exceeding the target link capacity at each control interval based on the expected traffic rates obtained by the prediction. In comparing the LTC with our proposed method, we intend to demonstrate that shortening the control interval reduces congestion despite the uncertainty of observed traffic.

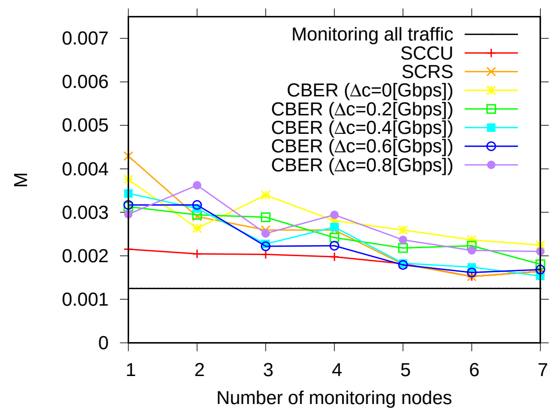

Fig. 3 Effect of stochastic control

(4) Monitoring all traffic

This method collects traffic data from all monitoring nodes at each time slot. Then, traffic is predicted, and routes are calculated in the same way as SCCU.

4.2 Results

Figure 3 shows the relationship between the number of nodes monitoring traffic at each time slot and the rate of congestion, where Δc is the same as in Eq. (24).

The results indicate that M depends on Δc in CBER. If Δc is small, the prediction error causes congestion because the expected traffic passing link l may be close to c_l. On the other hand, if Δc is large, we do not have sufficient resources to allocate to flows with large traffic. As a result, congestion may occur. In this evaluation, $\Delta c = 0.6$ achieves the smallest M. However, the optimal Δc depends on the patterns of traffic changes, and is difficult to be set in advance.

Figure 3 shows that SCRS achieves as value of M that is as small as it is with CBER with the optimal Δc when the number of monitoring nodes is larger than 5. SCRS considers probability distributions, and more resources are allocated to flows whose rates are large or uncertain. As a result, the probability that congestion will occur is reduced. However, as the number of monitoring nodes decreases, M increases because most of the flows that have a large impact on the performance become uncertain, which results in a lack of resources that can be allocated.

Our proposed SCCU achieved a low M value even when the number of monitoring nodes was 1. This is because SCCU selects monitoring nodes that can monitor the flows whose uncertainties significantly impact TE. As a result, SCCU accurately estimates the traffic rates of the flows.

SCCU can reconfigure the routes even when only partial traffic information is observed at each time slot. This allows for shorter control intervals. Thus, we evaluated the impact of shortening the control interval by comparing our method with LTC. For this evaluation, we obtained traffic information from n of 27 monitoring nodes every 10 s. In this case, SCRS and SCCU changed routes every 10 s, whereas LTC changed

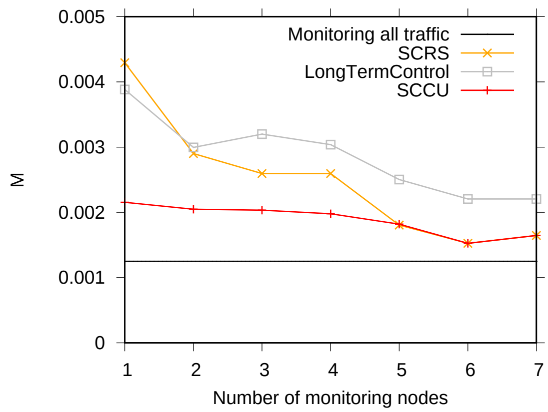

Fig. 4 Performance of conventionally proposed TE compared to our method

routes every $\frac{240}{n}$ s.

Figure 4 shows the rate of congestion achieved by each method. The figure indicates that SCRS and SCCU achieved a smaller rate of congestion than the LTC. That is, shortening the control interval reduced the rate of congestion. This is because a method with short intervals detects the risk of congestion and changes the routes so as to mitigate the risk soon after the risk is high. However, a method with large intervals cannot detect congestion and cannot mitigate congestion before the next time slot.

5. Scalability

In this section, we discuss scalability. As the size of the network increases, the calculation time may become prohibitively long. Thus, we first discuss computational complexity. Then, we propose approaches to accelerate the calculation.

5.1 Computational complexity

In each time slot, our method performs the following steps: estimation, prediction, route calculation, and monitoring node selection. Herein, we discuss the computational complexity of each step. We denote the number of nodes by n, number of flows by f, number of links by l, and number of candidate routes for each flow by k.

5.1.1 Estimation

Traffic is estimated by selecting previously estimated traffic or by monitored traffic for each flow. That is, the computational complexity of the estimation is $O(f)$.

5.1.2 Prediction

Our method predicts traffic by summing the vectors whose number of elements is the number of flows. That is, the computational complexity of the estimation is $O(f)$.

5.1.3 Route Calculation

The routes are calculated by repeating the updates using Eq. 19. That is, the calculation time depends on the number of updates.

Each update calculates the sum of the matrices whose number of elements is $f \cdot k$ That is, the computational complexity of each update is $O(f \cdot k)$.

5.1.4 Monitoring node selection

Our method selects monitoring nodes by repeating the selection of one monitoring node that minimizes $L^{\mathrm{opt}}(O_t)$. Each time we select a monitoring node, $L^{\mathrm{opt}}(O_t)$ is updated for all candidate monitoring nodes. That is, to select r monitoring noes from n candidates, the calculation of $L^{\mathrm{opt}}(O_t)$ is performed $\sum_{k=n-r+1}^{n} k$ times. Moreover, route calculation is required in order to calculate $L^{\mathrm{opt}}(O_t)$. Thus, the computational complexity is $O(f \cdot k) * O(\sum_{k=n-r+1}^{n} k) = O((\sum_{k=n-r+1}^{n} k) \cdot f \cdot k)$.

5.2 Acceleration

5.2.1 Acceleration of selecting monitoring node

According to the discussion in Section 5.1, with our method, selecting monitoring nodes requires the most calculation time. Therefore, we can accelerate the calculation time by accelerating the selection of monitoring nodes.

To do so, we focus on the flows passing the links whose probabilities of congestion are large, because a flow passing only the links whose probabilities of congestion are small has only a small impact on the performance. Even if the traffic rate of such a flow is uncertain, a route that avoids congestion can be calculated.

In addition, among the flows passing links with a high probability of congestion, we focus on flows with considerable uncertainty. This is because the impact of monitoring highly uncertain flows is larger than the impact of monitoring flows with less uncertainty.

Our method of acceleration selects monitoring nodes as follows, after predicting the traffic and calculating the routes:

1. Calculate the probability of congestion for each link by using the traffic predicted by the predictor and the routes calculated by the route controller.
2. Select as candidate flows those that pass links whose probabilities of congestion are larger than a threshold d.
3. Select one flow whose uncertainty, which is the defined by the variance obtained by the results of the prediction, is the highest among the candidate flows.
4. Select one monitoring node that can monitor the flows selected during Step 3, unless when selecting monitoring nodes, the cost $C(O_{t+1})$ does not exceed the threshold W. Otherwise, go to Step 7.

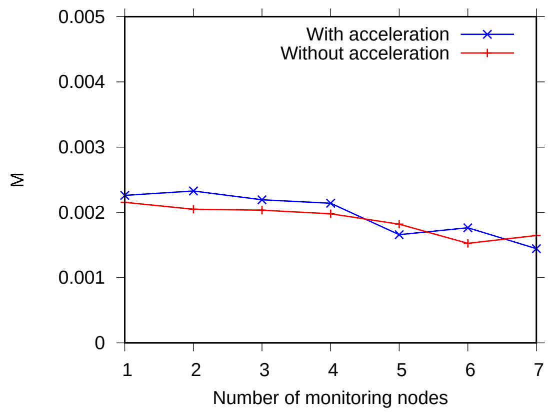

Fig. 5 Performance from selecting nodes based on predictive distributions

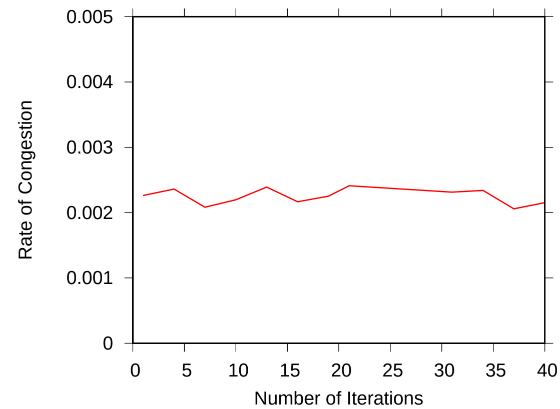

Fig. 6 Relationship between performance and the number of iterations

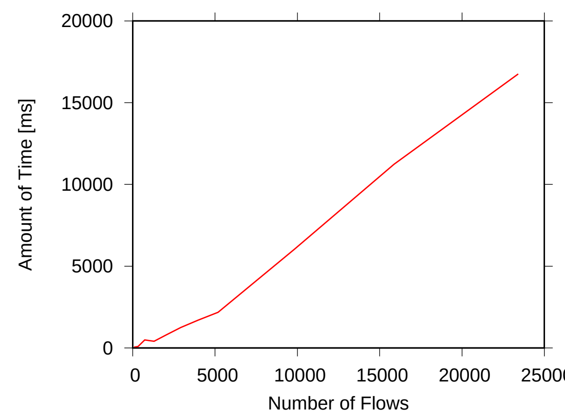

Fig. 7 Relationship between the number of flows and the calculation time

5. Eliminate from the candidate flows those that can be monitored by the monitoring node selected at Step 4.
6. Go to Step-3.
7. End.

The above steps do not require the calculation of $L^{\mathrm{opt}}(O_t)$. Thus, the computational complexity is merely $O(\sum_{k=n-r+1}^{n} k)$.

Our method of acceleration reduces the calculation time. However, it can also select different monitoring nodes without acceleration. Therefore, we evaluated the impact of the accelerated version.

As in Section 4, we used traffic trace data monitored by Internet2, and the network topology based on Internet2.

We set the number of monitoring nodes at each time period to 1. For simplicity, d, described in Section 5.2, was set to 0. The other simulation environments were set in the same manner as described in Section 4.1.

Figure 5 shows the rate of congested links achieved by our method with and without acceleration in the case of $s = 3$. The results indicate that acceleration does not significantly impact the rate of congested links.

This is because flows whose traffic rates are uncertain strongly impact the probabilities of congestion. As a result, the same monitoring nodes were selected by both methods.

5.2.2 Acceleration of route calculation

In addition to selecting monitoring nodes, calculating routes may require ample time. Indeed, the number of update iterations considerably impacts the calculation time. Therefore, we discuss the impact of the number of iterations on the rate of congested links.

Fig. 6 shows the impact of the number of iterations on the rate of congested links. The figure indicates that we can achieve a similar rate of congested links even when the number of iterations is set to 1. This is because we narrowed the control interval to only 10 s. As a result, significant route changes are not required during each time slot, and a sufficient solution can be found with only slight updates to the routes relative to the previous time slot.

5.2.3 Calculation time with acceleration

Finally, we evaluated the calculation time at each time slot for a large-scale network. To use larger network topologies, we connected s access routers to each PoP router. That is, the topology used in our evaluation had $9 * s$ access routers and 9 PoP routers.

As discussed in Section 5.1, the calculation time is a function of the number of flows. Therefore, we plotted the relation between the calculation time and the number of flows whose routes are configured. Figure 7 shows the calculation time of our method with acceleration, indicating that the calculation time is directly proportional to the number of flows. This is because the time needed to calculate route R occupies most of the calculation time, and it is proportional to the number of flows. The results thus indicate that our method with acceleration can complete the calculation for each time slot in 10 s for networks that include 15,000 flows.

If there are fewer flows that may cause congestion, we can accelerate our method even further, because SCCU calculates the uncertainty of each flow. By using this information, we can obtain flows that may cause congestion. Then, we update the routes such that only the routes of these flows are changed. By doing so, routing matrices that need to be updated are small, and the calculation time is proportional to the number of flows that may cause congestion.

6. Conclusion

In this paper, we proposed a framework for TE in cases where only partial traffic information can be obtained at each time slot. The framework was inspired by the decision-making process of the human brain. In our proposed framework, the controller (1) obtains a limited amount of traffic information, (2) estimates and predicts the probability distribution of traffic, (3) configures routes considering the probability distribution of future predicted traffic, and (4) selects traffic to monitor during the next period.

We discussed the details of the each step of our framework and its evaluation. The results demonstrate that our framework—in which TE and traffic monitoring cooperate—improves the performance of TE even when only partial traffic information is monitored during each time slot.

Our future work shall include the evaluation of our method in a larger, actual network. We shall also discuss the parameter settings for our method.

References

[1] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, K. Shiomoto, and T. Hashimoto, "Framework for traffic engineering under uncertain traffic information," Information and Communication Technology Convergence (ICTC), 2016 International Conference on, pp.264–266, IEEE, 2016.

[2] K. Satake, T. Otoshi, Y. Ohsita, and M. Murata, "Traffic engineering cooperating with traffic monitoring for the case with incomplete information," 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), pp.1–7, IEEE, 2017.

[3] E. Moreno, A. Beghelli, and F. Cugini, "Traffic engineering in segment routing networks," Computer Networks, vol.114, pp.23–31, 2017.

[4] S. Jeong, D. Lee, J. Hyun, J. Li, and J.W.K. Hong, "Application-aware traffic engineering in software-defined network," Network Operations and Management Symposium (APNOMS), 2017 19th Asia-Pacific, pp.315–318, IEEE, 2017.

[5] Y. Zhang and M. Moradi, "Sdn based interdomain and intradomain traffic engineering," July 4 2017. US Patent 9699116B2.

[6] M. Robinson, M. Milosavljevic, P. Kourtessis, S. Fisher, G.P. Stafford, J. Treiber, M.J. Burrell, and J.M. Senior, "Qoe based holistic traffic engineering in sdn enabled heterogeneous transport networks," Transparent Optical Networks (ICTON), 2017 19th International Conference on, pp.1–4, IEEE, 2017.

[7] M. Chiesa, G. Kindler, and M. Schapira, "Traffic engineering with equal-cost-multipath: An algorithmic perspective," IEEE/ACM Transactions on Networking, vol.25, no.2, pp.779–792, 2017.

[8] D. Sanvito, I. Filippini, A. Capone, S. Paris, and J. Leguay, "Adaptive robust traffic engineering in software defined networks," arXiv preprint arXiv:1712.05651, 2017.

[9] M. Katoh, I. Sato, and N. Watanabe, "Traffic engineering for iot," Information Networking (ICOIN), 2016 International Conference on, pp.195–200, IEEE, 2016.

[10] P. Kumar, Y. Yuan, C. Yu, N. Foster, R. Kleinberg, and R. Soulé, "Kulfi: Robust traffic engineering using semi-oblivious routing," arXiv preprint arXiv:1603.01203, 2016.

[11] M. Chiesa, G. Kindler, and M. Schapira, "Traffic engineering with equal-cost-multipath: An algorithmic perspective," IEEE/ACM Transactions on Networking, 2016.

[12] P. Sun, L. Vanbever, and J. Rexford, "Scalable programmable inbound traffic engineering," Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, p.12, ACM, 2015.

[13] H.H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter, "Traffic engineering with forward fault correction," ACM SIGCOMM Computer Communication Review, vol.44, no.4, pp.527–538, 2015.

[14] I.F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," Computer Networks, vol.71, pp.1–30, 2014.

[15] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine grained traffic engineering for data centers," Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies, p.8, ACM, 2011.

[16] D. Jiang, L. Nie, Z. Lv, and H. Song, "Spatio-Temporal Kronecker Compressive Sensing for Traffic Matrix Recovery," IEEE Access, vol.4, pp.3046–3053, 2016.

[17] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," ACM SIGCOMM Computer Communication Review, pp.267–278, ACM, 2009.

[18] S. Bitzer, J. Bruineberg, and S.J. Kiebel, "A Bayesian Attractor Model for Perceptual Decision Making," PLoS Comput Biol, vol.11, no.8, p.e1004442, 2015.

[19] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, K. Shiomoto, and T. Hashimoto, "Traffic engineering based on stochastic model predictive control for uncertain traffic change," Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, pp.1165–1170, IEEE, 2015.

Kodai Satake received an B.E. degree in engineering science in 2016 from Osaka University, where he is currently a postgraduate studying for a M.E. degree. His research interests include traffic engineering and traffic prediction. He is a student member of the IEEE.

Tatsuya Otoshi received an M.E. and Ph.D. degrees in information science and technology in 2017 and 2014 from Osaka University, where he is currently an specially appointed assistant professor in the Graduate School of Information Science and Technology. His research interests include traffic engineering and traffic prediction. He is a member of the IEEE.

Yuichi Ohsita received M.E. and Ph.D. degrees in information science and technology in 2005 and 2008 from Osaka University, where he is currently an assistant professor in the Graduate School of Information Science and Technology. His research interests include traffic matrix estimation and countermeasures against DDoS attacks. He is a member of IEICE, IEEE, and the Association for Computing Machinery (ACM).

Masayuki Murata received M.E. and Ph.D. degrees in information science and technology from Osaka University in 1984 and 1988. In April 1984, he joined the Tokyo Research Laboratory at IBM Japan as a researcher. From September 1987 to January 1989, he was an assistant professor with the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an associate

professor with the Graduate School of Engineering Science, Osaka University, and since April 1999, he has been a professor. He moved to the Graduate School of Information Science and Technology, Osaka University, in April 2004. He has published more than 300 papers in international and domestic journals and conferences. His research interests include computer communication networks, performance modeling, and evaluation. He is a fellow of IEICE and a member of IEEE, the Association for Computing Machinery (ACM), The Internet Society, and IPSJ.